

## PATENT ABSTRACTS OF JAPAN

(11) Publication number : 08-044788  
 (43) Date of publication of application : 16.02.1996

---

(51) Int.CI.

G06F 17/50  
 G01R 31/26  
 H01L 21/82

---

(21) Application number : 07-118712

(71) Applicant : TOSHIBA CORP

(22) Date of filing : 17.05.1995

(72) Inventor : SATOU HISATOMO

---

(30) Priority

Priority number : 06109862 Priority date : 24.05.1994 Priority country : JP

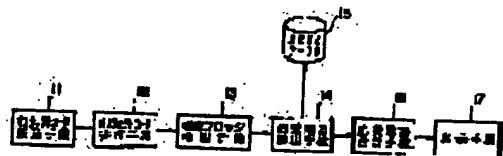
---

**(54) METHOD AND DEVICE FOR CALCULATING POWER CONSUMPTION OF INTEGRATED CIRCUIT**

**(57) Abstract:**

**PURPOSE:** To provide the method and device for calculating the power consumption of the integrated circuit which can optimize the power consumption and reduce the cost.

**CONSTITUTION:** This device is equipped with a function block detecting circuit 13 which detects function blocks required for specific operation among function blocks in order, a power consumption calculating means 14 which calculates the power consumption when the respective function blocks detected by the function block detecting means 13 operates, and a power consumption totalizing means 16 which totalizes the power consumption found by the function blocks through the power consumption calculating means 14 and outputs the sum as the power consumption of the whole integrated circuit.




---

**LEGAL STATUS**

[Date of request for examination] 26.01.2000

[Date of sending the examiner's decision of rejection] 26.02.2002

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision  
of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-44788

(43) 公開日 平成8年(1996)2月16日

(51) Int.Cl.  
G 06 F 17/50  
G 01 R 31/26  
H 01 L 21/82

識別記号 庁内整理番号

P I

技術表示箇所

G

9191-5H

G 06 F 15/60 654 Z  
H 01 L 21/82 C

審査請求 未請求 請求項の数 7 OL (全 12 頁)

(21) 出願番号 特願平7-118712

(71) 出願人 000003078

(22) 出願日 平成7年(1995)5月17日

株式会社東芝

(31) 優先権主張番号 特願平6-109862

神奈川県川崎市幸区堀川町72番地

(32) 優先日 平6(1994)5月24日

(72) 発明者 佐藤 寿一

(33) 優先権主張国 日本 (JP)

神奈川県川崎市幸区小向東芝町1 株式会

社東芝研究開発センター内

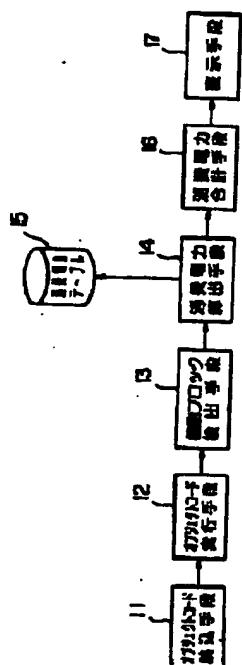
(74) 代理人 弁理士 佐藤 一雄 (外3名)

(54) 【発明の名称】 集積回路の消費電力算出方法及びその装置

(57) 【要約】

【目的】 消費電力の最適化及びコスト低減が可能な集積回路の消費電力算出方法及びその装置を提供する。

【構成】 機能ブロックのうち所定の動作に必要なものを順次検出する機能ブロック検出手段13と、この機能ブロック検出手段13により検出されたそれぞれの機能ブロックが動作するときの消費電力を算出する消費電力算出手段14と、この消費電力算出手段14がそれぞれの機能ブロック毎に求めた消費電力を合計し、集積回路全体の消費電力として出力する消費電力合計手段16とを備える。



(2)

1

## 【特許請求の範囲】

【請求項1】複数の機能ブロックを有する集積回路の消費電力を算出する方法において、

前記機能ブロックのうち、所定の動作に必要なものを順次検出していくステップと、

検出された前記機能ブロックが動作するときのそれぞれの消費電力を求めるステップと、

前記機能ブロック毎に求めた消費電力を合計し、集積回路全体の消費電力として出力するステップとを備えたことを特徴とする集積回路の消費電力算出方法。

【請求項2】前記消費電力を求めるステップでは、前記機能ブロック毎に消費電力が予め設定されてテーブルに登録されており、前記テーブルを参照することで検出された前記機能ブロックの消費電力を求ることを特徴とする請求項1記載の集積回路の消費電力算出方法。

【請求項3】前記消費電力を求めるステップでは、検出された前記機能ブロックにn (nは1以上の整数)個のベクトルAi (iは1以上でn以下の整数)が入力される場合、所定の関数f及び所定の定数constにより

$$P = f (A_1, A_2, \dots, A_n) + \text{const}$$

で表される近似式を用いて消費電力Pを算出することを特徴とする請求項1記載の集積回路の消費電力算出方法。

【請求項4】前記消費電力を求めるステップでは、検出された前記機能ブロックにn個のベクトルAiが入力され、検出された前記機能ブロックからm (mは1以上の整数)個のベクトルCj (jは1以上でm以下の整数)が出力される場合、所定の関数f及び所定の定数constにより

$$P = f (A_1, A_2, \dots, A_n, C_1, C_2, \dots, C_m) + \text{const}$$

で表される近似式を用いて消費電力Pを算出することを特徴とする請求項1記載の集積回路の消費電力算出方法。

【請求項5】前記消費電力を求めるステップでは、検出された前記機能ブロックにn個のベクトルAiが入力される場合、現時刻のAiをAi (t)、前時刻のAi (t-1)で表す時、所定の関数f及び所定の定数constにより

$$P = f (A_1 (t), A_1 (t-1), A_2 (t), A_2 (t-1), \dots, A_n (t), A_n (t-1)) + \text{const}$$

で表される近似式を用いて消費電力Pを算出することを特徴とする請求項1記載の集積回路の消費電力算出方法。

【請求項6】前記消費電力を求めるステップでは、検出された前記機能ブロックにn個のベクトルAiが入力され、検出された前記機能ブロックからm個のベクトルCj (j=1, 2, ..., m)が出力される場合、現時刻のAiをAi (t)、前時刻のAiをAi (t-1)で表す時

し、現時刻のCjをCj (t)、前時刻のCjをCj (t-1)で表す時、所定の関数f及び所定の定数constにより

$$P = f (A_1 (t), A_1 (t-1), A_2 (t), A_2 (t-1), \dots, A_n (t), A_n (t-1), C_1 (t), C_1 (t-1), C_2 (t), C_2 (t-1), \dots, C_m (t), C_m (t-1)) + \text{const}$$

で表される近似式を用いて消費電力Pを算出することを特徴とする請求項1記載の集積回路の消費電力算出方法。

【請求項7】複数の機能ブロックを有する集積回路の消費電力を算出する装置において、

前記機能ブロックのうち、所定の動作に必要なものを順次検出していく機能ブロック検出手段と、

前記機能ブロック検出手段により検出されたそれぞれの前記機能ブロックが動作するときのそれぞれの消費電力を算出する消費電力算出手段と、

前記消費電力算出手段がそれぞれの前記機能ブロック毎に求めた消費電力を合計し、集積回路全体の消費電力をとして出力する消費電力合計手段とを備えたことを特徴とする集積回路の消費電力算出装置。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】本発明は、集積回路の消費電力を算出する方法及びその装置に関する。

## 【0002】

【従来の技術】集積回路の設計は、例えば仕様設計工程、アーキテクチャ設計工程、機能設計工程、論理設計工程、回路設計工程、レイアウト設計工程というように、上位の仕様設計工程から下位のレイアウト設計工程へ向けて進められる。ここで、アーキテクチャ設計工程では、例えば図3に示されるような各機能ブロック21～34間のデータの流れが設計される。論理設計工程では、図4に示されたNANDゲート41及び42を用いた論理回路のような回路設計が行われる。回路設計工程では、図5に具体的にトランジスタ51～54として示されたようなトランジスタ回路の設計が行われる。

【0003】消費電力P (W) の算出は、電流をI (A)、動作電圧をE (V) とした場合に  $P = I E$  で表される式を用いて行われる。従来は、図4に示されたような論理回路、あるいは図5に示されたようなトランジスタ回路を用いて回路シミュレーションを行い、電流及び電圧を計算して消費電力Pを求めていた。図5に示されたトランジスタ回路を例にとると、電源電圧VDD端子から接地電圧Vs端子へ向けて流れる電流Iを、計算機による回路シミュレーションで求めて、電源電圧VDDと電流Iとから消費電力Pを算出することができる。

## 【0004】

【発明が解決しようとする課題】ところで、消費電力に大きな影響を与えるのは上位の設計工程である。例え

ば、アーキテクチャ設計工程における設計の変更は、最終的な消費電力に大きく影響する。このため、アーキテクチャ設計工程で消費電力を見積もることは、消費電力の低減を図る上で極めて有効である。

【0005】しかし、上述したように従来は論理回路又はトランジスタ回路を用いて消費電力の算出を行っていたため、論理回路設計又はトランジスタ回路設計といった下位の設計工程まで進まなければ消費電力の見積もりは不可能であった。そこで、アーキテクチャ設計工程において消費電力の最適化を図ることをあきらめて、論理設計工程、あるいは回路設計工程で消費電力の最適化を行っていた。この場合には、下位の設計工程のみにおける最適化であるため、消費電力の低減を図る上であまり有効な方法ではなかった。

【0006】あるいは、論理回路設計工程又はトランジスタ回路設計工程において消費電力を算出し、この後アーキテクチャ設計工程へフィードバックさせてアーキテクチャの再設計を行い、さらにこのアーキテクチャに基づいた論理回路及びトランジスタ回路を設計し直す場合もあった。この場合には、設計工数及び時間が増加しこストの上昇を招いていた。

【0007】本発明は上記事情に鑑みてなされたもので、消費電力の最適化及びコスト低減が可能な集積回路の消費電力算出方法及びその装置を提供することを目的とする。

【0008】

【課題を解決するための手段】本発明の集積回路の消費電力算出方法は、複数の機能ブロックを有する集積回路の消費電力を算出する方法であって、前記機能ブロックのうち、所定の動作に必要なものを順次検出していくステップと、検出された前記機能ブロックが動作するときのそれぞれの消費電力を求めるステップと、前記機能ブロック毎に求めた消費電力を合計し、集積回路全体の消費電力として出力するステップとを備えたことを特徴としている。

【0009】ここで、前記消費電力を求めるステップでは、前記機能ブロック毎に消費電力が予め設定されてテーブルに登録されており、前記テーブルを参照することで検出された前記機能ブロックの消費電力を求めてよい。

【0010】また、前記消費電力を求めるステップでは、次のような近似式を用いて消費電力Pを算出してもよい。

(1) 検出された前記機能ブロックにn個のベクトルAiが入力される場合、所定の関数f及び所定の定数constにより

$P = f (A_1, A_2, \dots, A_n) + \text{const}$   
で表される近似式。

(2) 検出された前記機能ブロックにn個のベクトルAiが入力される場合、任意の個数選ばれたAiによつ

て表される所定の関数f及び所定の定数constにより  
 $P = f (A_1, \dots) + \text{const}$   
で表される近似式。

(3) 検出された前記機能ブロックにn個のベクトルAiが入力される場合、前記ベクトルAiのビット幅をWAi、前記ベクトルAiのうち所定の論理値にあるもののビット数をHAI、前記ベクトルAに起因して消費される電力の最大値をPAiとし、所定の関数f及び所定の定数constにより

10  $P = f (PA_1, PA_2, \dots, PA_n, HA_1, HA_2, \dots, HA_n, WA_1, WA_2, \dots, WA_n) + \text{const}$   
で表される近似式。

(4) 前記消費電力を求めるステップにおいて用いられる関数fが、

$P = f = PA_1 \cdot HA_1 / WA_1 + PA_2 \cdot HA_2 / WA_2 + \dots + PA_n \cdot HA_n / WA_n + \text{const}$   
で表される近似式。

(5) 検出された前記機能ブロックにn個のベクトルAiが入力される場合、前記ベクトルAiのビット幅をWAi、前記ベクトルAiのうち所定の論理値にあるもののビット数をHAI、前記ベクトルAに起因して消費される電力の最大値をPAiとし、任意の個数選ばれたAiによって表される所定の関数f及び所定の定数constにより

$P = f (PA_i, HA_i, WA_i, \dots) + \text{const}$   
で表される近似式。

(6) 前記消費電力を求めるステップにおいて用いられる関数fが、

$P = f = PA_i \cdot HA_i / WA_i + \dots + \text{const}$   
で表される近似式。

(7) 検出された前記機能ブロックにn個のベクトルAiが入力され、検出された前記機能ブロックからm個のベクトルCjが出力される場合、所定の関数f及び所定の定数constにより

$P = f (\Lambda_1, \Lambda_2, \dots, \Lambda_n, C_1, C_2, \dots, C_m) + \text{const}$   
で表される近似式。

(8) 検出された前記機能ブロックにn個のベクトルAiが入力され、検出された前記機能ブロックからm個のベクトルCjが出力される場合、任意の個数選ばれたAiと任意の個数選ばれたCjによって表される所定の関数f及び所定の定数constにより

$P = f (A_i, \dots, C_j, \dots) + \text{const}$   
で表される近似式。

(9) 検出された前記機能ブロックにn個のベクトルAiが入力され、検出された前記機能ブロックからm個のベクトルCjが出力される場合、前記ベクトルAiのうち所定の論理値にあるもののビット数をHAI、前記ベクトルAに起因して消費される電力の最大値をPAiとし、前記ベク

トルC<sub>j</sub> のビット幅をWC<sub>j</sub> 、前記ベクトルC<sub>i</sub> のうち所定の論理値にあるもののビット数をHC<sub>i</sub> 、前記ベクトルCに起因して消費される電力の最大値をPC<sub>i</sub> とし、所定の関数f 及び所定の定数const により

$$P = f (PA_1, PA_2, \dots, PA_n, HA_1, HA_2, \dots, HA_n, WA_1, WA_2, \dots, WA_n, PC_1, PC_2, \dots, PC_m, HC_1, HC_2, \dots, HC_m, WC_1, WC_2, \dots, WC_m) + \text{const}$$

で表される近似式。

(10) 前記消費電力を求めるステップで用いられる関数f が、

$$P = f = PA_1 \cdot HA_1 / WA_1 + PA_2 \cdot HA_2 / WA_2 + \dots + PA_n \cdot HA_n / WA_n + PC_1 \cdot HC_1 / WC_1 + PC_2 \cdot HC_2 / WC_2 + \dots + PC_m \cdot HC_m / WC_m + \text{const}$$

で表される近似式。

(11) 検出された前記機能ブロックにn個のベクトルAi が入力され、検出された前記機能ブロックからm個のベクトルCj が出力される場合、前記ベクトルAi のビット幅をWAi 、前記ベクトルAi のうち所定の論理値にあるもののビット数をHAi 、前記ベクトルAi に起因して消費される電力の最大値をPAi とし、前記ベクトルCj のビット幅をWCj 、前記ベクトルCi のうち所定の論理値にあるもののビット数をHCi 、前記ベクトルCに起因して消費される電力の最大値をPCi とし、任意の個数選ばれたAi と任意の個数選ばれたCj によって表される所定の関数f 及び所定の定数により

$$P = f (PA_i, HA_i, WA_i, \dots, PC_j, HC_j, WC_j, \dots) + \text{const}$$

で表される近似式。

(12) 前記消費電力を求めるステップで用いられる関数f が、

$$P = f = PA_1 \cdot HA_1 / WA_1 + \dots + PC_j \cdot HC_j / WC_j + \dots + \text{const}$$

で表される近似式。

(13) 検出された前記機能ブロックにn個のベクトルAi が入力される場合、前記ベクトルAi のビット幅をWAi 、前記ベクトルAのうち前時刻から現在の時刻へ移行する際に論理値が変化したもののビット数をHAi 、前記ベクトルAに起因して消費される電力の最大値をPAi とし、所定の関数f 及び所定の定数により

$$P = f (PA_1, PA_2, \dots, PA_n, HA_1, HA_2, \dots, HA_n, WA_1, WA_2, \dots, WA_n) + \text{const}$$

で表される近似式。

(14) 前記消費電力を求めるステップで用いられる関数f が、

$$P = f = PA_1 \cdot HA_1 / WA_1 + PA_2 \cdot HA_2 / WA_2 + \dots + PA_n \cdot HA_n / WA_n + \text{const}$$

で表される近似式。

(15) 検出された前記機能ブロックにn個のベクト

ルAi が入力される場合、前記ベクトルAi のビット幅をWAi 、前記ベクトルAのうち前時刻から現在の時刻へ移行する際に論理値が変化したもののビット数をHAI<sub>i</sub> 、前記ベクトルAに起因して消費される電力の最大値をPAi とし、任意の個数選ばれたAi によって表される所定の関数f 及び所定の定数により  $P = f (PA_i, HA_i, WA_i \dots) + \text{const}$  で表される近似式。

(16) 前記消費電力を求めるステップで用いられる関数f が、

$$P = f = PA_1 \cdot HA_1 / WA_1 + \dots + \text{const}$$

で表される近似式。

(17) 検出された前記機能ブロックにn個のベクトルAi が入力され、検出された前記機能ブロックからm個のベクトルCj が出力される場合、前記ベクトルAi のビット幅をWAi 、前記ベクトルAのうち前時刻から現在の時刻へ移行する際に論理値が変化したもののビット数をHAI<sub>i</sub> 、前記ベクトルAに起因して消費される電力の最大値をPAi とし、前記ベクトルCj のビット幅をWCj 、前記ベクトルCのうち前時刻から現在の時刻へ移行する際に論理値が変化したもののビット数をHIC<sub>i</sub> 、前記ベクトルCに起因して消費される電力の最大値をPCi とし、所定の関数f 及び所定の定数const により

$$P = f (PA_1, PA_2, \dots, PA_n, HA_1, HA_2, \dots, HA_n, WA_1, WA_2, \dots, WA_n, PC_1, PC_2, \dots, PC_m, HC_1, HC_2, \dots, HC_m, WC_1, WC_2, \dots, WC_m) + \text{const}$$

で表される近似式。

(18) 前記消費電力を求めるステップで用いられる関数f が、

$$P = f = PA_1 \cdot HA_1 / WA_1 + PA_2 \cdot HA_2 / WA_2 + \dots + PA_n \cdot HA_n / WA_n + PC_1 \cdot HC_1 / WC_1 + PC_2 \cdot HC_2 / WC_2 + \dots + PC_m \cdot HC_m / WC_m + \text{const}$$

で表される近似式。

(19) 検出された前記機能ブロックにn個のベクトルAi が入力され、検出された前記機能ブロックからm個のベクトルCj が出力される場合、前記ベクトルAi のビット幅をWAi 、前記ベクトルAのうち前時刻から現在の時刻へ移行する際に論理値が変化したもののビット数をHAI<sub>i</sub> 、前記ベクトルAに起因して消費される電力の最大値をPAi とし、前記ベクトルCj のビット幅をWCj 、前記ベクトルCのうち前時刻から現在の時刻へ移行する際に論理値が変化したもののビット数をHIC<sub>i</sub> 、前記ベクトルCに起因して消費される電力の最大値をPCi とし、任意の個数選ばれたAi と任意の個数選ばれたCj によって表される所定の関数f 及び所定の定数const により

$$P = f (PA_i, HA_i, WA_i, \dots, PC_j, HC_j, WC_j, \dots) + \text{const}$$

で表される近似式。

(20) 前記消費電力を求めるステップで用いられる関数  $f$  が、

$$P - f = PA_i \cdot HA_i / WA_i + \dots + C_j \cdot HC_j / WC_j + \dots + const$$

で表される近似式。

(21) 検出された前記機能ブロックに  $n$  個のベクトル  $A_i$  が入力される場合、現時刻の  $A_i$  を  $A_i(t)$ 、前時刻の  $A_i(t-1)$  で表す時、所定の関数  $f$  及び所定の定数  $const$  により

$$P = f(A_1(t), A_1(t-1), A_2(t), A_2(t-1), \dots, A_n(t), A_n(t-1)) + const$$

で表される近似式。

(22) 検出された前記機能ブロックに  $n$  個のベクトル  $A_i$  が入力され、検出された前記機能ブロックから  $m$  個のベクトル  $C_j$  が outputされる場合、現時刻の  $A_i$  を  $A_i(t)$ 、前時刻の  $A_i(t-1)$  で表し、現時刻の  $C_i$  を  $C_i(t)$ 、現時刻の  $C_i(t-1)$  で表す時、所定の関数  $f$  及び所定の定数  $const$  により

$$P = f(A_1(t), A_1(t-1), A_2(t), A_2(t-1), \dots, A_n(t), A_n(t-1), C_1(t), C_1(t-1), C_2(t), C_2(t-1), \dots, C_m(t), C_m(t-1)) + const$$

で表される近似式。

【0011】本発明の消費電力算出装置は、複数の機能ブロックを有する集積回路の消費電力を算出する装置であって、前記機能ブロックのうち、所定の動作に必要なものを順次検出する機能ブロック検出手段と、前記機能ブロック検出手段により検出されたそれぞれの前記機能ブロックが動作するときの消費電力を算出する消費電力算出手段と、前記消費電力算出手段がそれぞれの前記機能ブロック毎に求めた消費電力を合計し、集積回路全体の消費電力として出力する消費電力合計手段とを備えたことを特徴としている。

【0012】前記機能ブロックがそれぞれ動作した場合に消費される電力を予め登録したテーブルをさらに備え、前記消費電力算出手段は前記テーブルを参照することで、検出された当該機能ブロックの消費電力を求めてよい。

【0013】消費電力算出装置においても、上述したいずれかの近似式を用いて消費電力を求めてよい。

【0014】

【作用】集積回路を複数の機能ブロックで構成することは上位の設計工程において可能であり、この段階で動作に必要な機能ブロックを検出して消費電力を求め、合計することで回路全体の消費電力を算出することができる。このように上位の設計工程で消費電力を見積もることで、消費電力の最適化が可能となると同時に、下位の

設計工程で消費電力を見積もる場合よりも設計時間が短縮されコストが低減される。

【0015】

【実施例】以下、本発明の一実施例について図面を参照して説明する。本実施例では、マイクロプロセッサの設計において消費電力の算出を行う場合に本発明を適用したものに相当する。

【0016】図1に、本実施例による集積回路の消費電力算出装置の構成を示す。本装置は、オブジェクトコード読込手段11、オブジェクトコード実行手段12、機能ブロック検出手段13、消費電力算出手段14、消費電力テーブル15、消費電力合計手段16、及び表示手段17を備えている。このような装置を用いて、消費電力の算出を行う方法について、図2のフローチャートを用いて説明する。

【0017】ステップ101として、図示されていないコンパイラがソースコードを与えられてコンパイルを行い生成したオブジェクトコードをオブジェクトコード読込手段11が読み込み、オブジェクトコード実行手段12に与える。

【0018】オブジェクトコード実行手段12は、ステップ102として与えられたオブジェクトコードを解析して命令を取り出し、実行する。

【0019】ステップ103として、マイクロプロセッサを構成する複数の機能ブロックのうち、命令を実行するために必要な機能ブロックを、機能ブロック検出手段13が検出する。

【0020】以下の式(1)～(3)に表されたアセンブリコードで表現された命令を実行する場合を例にとり説明する。

ADD r3, r1, r2 … (1)

ADD r6, r4, r5 … (2)

ADD r9, r7, r8 … (3)

式(1)～(3)でそれぞれ表された命令(1)～(3)を、パイプラインを用いて実行するとき、図3に示された機能ブロック21～34のうち必要なものは次のようにして検出される。

1. (時刻t1) 先ず、命令(1)をフェッチする。機能ブロック検出手段13は、フェッチに必要な機能ブロックとして命令キャッシュ22を検出する。さらに、次時刻におけるアドレスを決定するために必要な機能ブロックとして、インクリメンタ23を検出する。

2. (時刻t2) 命令(1)をデコードし、レジスタr1及びr2からデータを読み出すために必要な機能ブロックとして、レジスタファイル26を検出する。

【0021】同時に、命令(2)のフェッチが開始される。命令(2)のフェッチから次時刻のアドレス決定の動作までは命令(1)の場合と同様であり、順に命令キャッシュ22、インクリメンタ23を機能ブロック検出手段13が検出する。

3. (時刻 t 3) 命令 (1) を実行する。レジスター 1 のデータとレジスター 2 のデータとを加算する。この加算動作に必要な機能ブロックとして、算術論理演算器 28 を機能ブロック検出手段 13 が検出する。

【0022】同時に、命令 (2) をデコードし、レジスター 4 及び 5 からデータを読み出すために必要な機能ブロックとして、レジスタファイル 26 を検出する。

【0023】さらに同時に、命令 (3) のフェッチが同時に行われる。命令 (3) のフェッチから次時刻のアドレス決定の動作までは、命令 (1) 又は命令 (2) の場合と同様であり、順に命令キャッシュ 22、インクリメント 23 を機能ブロック検出手段 13 が検出する。

4. (時刻 t 4) 命令 (2) を実行するため、レジスター 4 のデータとレジスター 5 のデータとを加算するが、この動作に必要な機能ブロックとして算術論理演算器 28 を機能ブロック検出手段 13 が検出する。

【0024】また、同時に命令 (3) のデコードに必要な機能ブロックとして、レジスター 7 のデータとレジスター 8 のデータとを読み出すためにレジスタファイル 26 を検出する。

【0025】このようにして、それぞれの動作に必要な機能ブロックを順次検出していく。

【0026】ステップ 104 として、検出された機能ブロックの消費電力を消費電力算出手段 14 がそれぞれ求める。ここで、消費電力テーブル 15 には予め各機能ブロックがそれぞれ動作した場合に消費される電力が登録されており、消費電力算出手段 14 が消費電力テーブル 15 を参照することで機能ブロックの消費電力を求める。ところで、本実施例における消費電力の算出は下位の設計工程まで進んでいない段階で行う。このため、各機能ブロックの論理回路構成又はトランジスタ回路構成はまだ決定されておらず、消費電力の設定は経験に基いて行う必要がある。

【0027】ステップ 105 として、各機能ブロック毎に求めた消費電力を消費電力合計手段 16 が合計し、回\*

$$P = PA \cdot CA / WA + PB \cdot CB / WB \quad \dots \quad (4)$$

ここで、CA は入力ベクトル A のうち、前時刻から現在の時刻へ移行する際に値が変化したもののビット数、CB は入力ベクトル B のうち、前時刻から現在の時刻へ移行する際に値が変化したもののビット数である。

【0037】ここで、近似式 (4) と (5) とを比較した場合、以下のような相違がある。近似式 (4) を用いた場合は、入力ベクトルの現在の論理値のみを用いて演算を行うことができ、論理値の時間的な変化を考慮する必要がない。このため、演算が簡易であり高速に処理することができる。また、演算を行う際に必要なメモリの容量も小さくてよい。

※

$$P = PA \cdot CA / WA + PB \cdot CB / WB + PC \cdot CC / WC \quad \dots \quad (5)$$

ここで、WC は出力ベクトル C のビット幅、CC は入力ベクトル C のうち、前時刻から現在の時刻へ移行する際

10 \* 路全体の消費電力を算出する。

【0028】ステップ 106 として、実行すべき命令が存在するか否かを判断する。命令が存在する場合には、ステップ 102 からステップ 105 までの処理を繰り返し、命令が存在しない場合はステップ 107 へ進む。

【0029】ステップ 107 では、最終的な消費電力の値を表示手段 17 が表示する。

【0030】消費電力値の表示は、例えば図 6 に示されたように時間に対する消費電力の変化という形で表示してもよい。

【0031】消費電力算出手段 14 において消費電力を算出する場合、上述した実施例では予め機能ブロック毎に見積もった消費電力をテーブルに格納しておき、このテーブルを参照することで消費電力を求めていた。

【0032】これに対し、テーブルを用いずに次のような近似式を用いて消費電力の算出を行ってもよい。

【0033】例えば、ある動作に必要な機能ブロックが算術論理演算器 30 である場合、この算術論理演算器 30 には二つのベクトル A 及び B が入力されるため、次の近似式 (4) を用いててもよい。

$$P = PA \cdot HA / WA + PB \cdot HB / WB \quad \dots \quad (4)$$

ここで、WA は入力ベクトル A のビット幅、WB は入力ベクトル B のビット幅、HA は入力ベクトル A のうち、論理「1」であるもののビット数、HB は入力ベクトル B のうち、論理「1」であるもののビット数、PA は入力ベクトル A の全てのビット数が論理「1」であるときの入力ベクトル A の入力に起因する消費電力、PB は入力ベクトル B の全てのビット数が論理「1」であるときの入力ベクトル B の入力に起因する消費電力であるとする。

【0035】あるいは、近似式 (4) の替わりに次の近似式 (5) を用いて消費電力を求めてよい。

$$P = PA \cdot CA / WA + PB \cdot CB / WB + PC \cdot CC / WC \quad \dots \quad (5)$$

※ 【0038】逆に、近似式 (5) を用いた場合は、入力ベクトルの論理値が変化するときに消費する電力を求めることができるため、CMOS 回路における消費電力の算出に好適である。CMOS 回路では、入力ベクトルが論理値「1」であることよりも、論理値が変化するときに流れる充放電電流の影響の方が消費電力全体に占める割合が大きいからである。

【0039】あるいは、近似式 (5) の替わりに次の近似式 (6) を用いて消費電力を求めてよい。

$$P = PA \cdot CA / WA + PB \cdot CB / WB + PC \cdot CC / WC \quad \dots \quad (6)$$

ここで、WC は出力ベクトル C のビット幅、CC は入力ベクトル C のうち、前時刻から現在の時刻へ移行する際

に値が変化したもののビット数、PC は入力ベクトル C に起因する消費電力の最大値であるとする。

【0041】ここで、近似式(5)と(6)とを比較した場合、以下のような相違がある。算術論理演算器30で加算を行う場合を考える。

【0042】前時刻のベクトルA、B、Cがそれぞれ0111、0000、0111で、現時刻のベクトルA、B、Cがそれぞれ0111、0001、1000の場合は、WA、WB、WCは全て4、CA、CB、CCはそれぞれ0、1、4であり、近似式(5)によれば $P = P_B + 4 P_C$ より求まる。これを事例1とする。

【0043】前時刻のベクトルA、B、Cがそれぞれ0110、0000、0110で、現時刻のベクトルA、B、Cがそれぞれ0110、0001、0111の場合は、WA、WB、WCは全て4、CA、CB、CCはそれぞれ0、1、1であり、近似式(5)によれば $P = P_B + P_C$ より求まる。これを事例2とする。

【0044】事例1と2を比較すると、近似式(5)によれば算術論理演算器13の消費電力は同じである。しかし、事例1の場合はキャリー伝搬が発生しているため、算術論理演算器13の回路中の動作部分が大きく、事例2に比べて消費電力も大きいはずである。近似式(6)を用いれば、事例1の算術論理演算器13の消費電力は、事例2のそれよりも大きいことが求められ、より精度の高い消費電力を算出できる。

【0045】シミュレーションにより近似式(5)と(6)の精度の比較を行った結果を示す。

【0046】本実施例による算出方法に基づく近似式を用いると、ベクトルの変化のみをもとに算出するので、実際の回路では異なる電力を消費するような演算に対しても、同じ消費電力が求まることになる。例えば、実際の回路では $(2^4)^2 = 256$ 通りの消費電力が存在するが、近似式(5)を用いる場合は $5 \times 5 = 25$ 通り、近似式(6)を用いる場合は $5 \times 5 \times 5 = 125$ 通りの答えしか存在しないことになる。

【0047】近似式の精度を表すのは以下の式(7)で定義される分散を求めればよい。分散とはサンプル値のバラツキを示す値で、値が小さいほどサンプル値のバラツキが小さい。

【0048】

$$V(X) = \sum (x - u)^2 P(x) \quad \dots \dots (7)$$

ここで $V(X)$ は集合Xの分散、 $x$ は集合Xの要素、 $u$ は集合Xの平均、 $P(x)$ は $x$ の起こる確率である。

【0049】したがって、近似式(5)あるいは(6)を用いた場合に同じ消費電力が得られる異なるベクトルを用いた時に、実際の回路での消費電力の分散が小さければ、近似式(5)あるいは(6)は、精度の高い近似式であることになる。

10 【0050】まず、算術論理演算器の回路には4ビットのリップル・キャリー・アダーセを選んだ。

【0051】出力を考慮しない近似式(5)を用いた $5 \times 5 = 25$ 通りの消費電力の値に対して、出力を考慮する近似式(6)を用いた消費電力は $5 \times 5 \times 5 = 125$ 通りであり、前者の1つに対して5通りが対応するが、前者の分散と後者のそれを比較すると、後者の125通りのうち、前者の分散よりも大きくなつたのは23通りであった。その分散の値は、前者と比べて最悪の場合で1.3倍である。

20 【0052】逆に出力を考慮しない場合の分散は、最悪の場合、出力を考慮した場合の51.0倍であった。

【0053】統いて、算術論理演算器の回路に4ビットのキャリー・ルックアヘッド・アダーセを選んだ場合のシミュレーション結果を紹介する。

【0054】出力を考慮しない近似式(5)を用いた $5 \times 5 = 25$ 通りの消費電力の値に対して、出力を考慮する近似式(6)を用いた消費電力は $5 \times 5 \times 5 = 125$ 通りであり、前者の1つに対して5通りが対応するが、前者の分散と後者のそれを比較すると、後者の125通りのうち、前者の分散よりも大きくなつたのは20通りであった。その分散の値は、前者と比べて最悪の場合で1.8倍である。

【0055】逆に出力を考慮しない場合の分散は、最悪の場合、出力を考慮した場合の9.0倍であった。

【0056】入力Aの変化量が4の場合の分散を以下に示す。下線を引いた分散が、出力を考慮しない場合よりも大きな値となつた分散である。

【0057】

【表1】

B \ C	考慮なし	0	1	2	3	4
0	$1.8 \times 10^{-7}$		$2.0 \times 10^{-8}$	$2.6 \times 10^{-8}$	$2.4 \times 10^{-8}$	$2.0 \times 10^{-8}$
1	$1.7 \times 10^{-7}$	$3.0 \times 10^{-8}$	$5.8 \times 10^{-8}$	$1.6 \times 10^{-7}$	$2.5 \times 10^{-7}$	$3.1 \times 10^{-8}$
2	$2.1 \times 10^{-7}$	$7.4 \times 10^{-8}$	$1.6 \times 10^{-7}$	$3.1 \times 10^{-7}$	$1.6 \times 10^{-7}$	$5.0 \times 10^{-8}$
3	$2.8 \times 10^{-7}$	$1.6 \times 10^{-7}$	$4.6 \times 10^{-7}$	$1.7 \times 10^{-7}$	$5.8 \times 10^{-8}$	$1.9 \times 10^{-8}$
4	$1.4 \times 10^{-7}$	$6.0 \times 10^{-8}$	$2.3 \times 10^{-8}$	$5.5 \times 10^{-8}$	$1.1 \times 10^{-7}$	

上の表で、左の欄から順に、Bの変化量、出力Cの考慮なしの場合の分散、出力Cを考慮した場合の分散であつて、出力Cの変化量が0、1、2、3、4の場合の分散を表す。

【0058】以上のシミュレーション結果から、出力を考慮する(6)式の方が、消費電力算出の精度が高いことがわかる。

【0059】以上、二つの入力ベクトルAおよびBを用いて表した近似式(4)、前時刻から現在の時刻へ移行する際に値が変化したベクトルを用いて表した近似式\*

$$P = f(A_1, A_2, \dots, A_n) + \text{const} \quad \dots \quad (8)$$

あるいは、機能ブロックに入力されるすべてのベクトルを用いるのではなく、任意の個数選ばれたベクトルA<sub>i</sub>を用いて表わされる関数fによる次の(9)式を用いてよい。

【0062】

$$P = f(A_1, \dots) + \text{const} \quad \dots \quad (9) \quad \text{※}$$

$$P = f(P_{A1}, P_{A2}, \dots, P_{An}, H_{A1}, H_{A2}, \dots, H_{An}, W_{A1}, \dots, W_{An}) + \text{const} \quad \dots \quad (10)$$

で表わされる近似式を用いて消費電力を求めることもできる。

【0063】ここで、上記(10)式は次の(11)式★

$$P = f = P_{A1} \cdot H_{A1} / W_{A1} + P_{A2} \cdot H_{A2} / W_{A2} + \dots + P_{An} \cdot H_{An} / W_{An} + \text{const} \quad \dots \quad (11)$$

$$P = f = P_{A1} \cdot H_{A1}^2 / W_{A1} + P_{A2} \cdot H_{A2}^2 / W_{A2} + \dots + P_{An} \cdot H_{An}^2 / W_{An} + \text{const} \quad \dots \quad (12)$$

この(12)式は、機能ブロックが例えばアレイ構造をした乗算器である場合には、有効である。

【0065】上記(10)式では、機能ブロックに入力される入力ベクトルAの全てを用いているが、次の(1)式

$$P = f = P_{Ai} \cdot H_{Ai} / W_{Ai} + \dots + \text{const} \quad \dots \quad (13)$$

この(13)式においても(14)式のような一次線形式のみならず、例えば(15)式のような二次非線形式◆も含まれる。

$$P = f = P_{Ai} \cdot H_{Ai}^2 / W_{Ai} + \dots + \text{const} \quad \dots \quad (14)$$

$$P = f = P_{Ai} \cdot H_{Ai}^3 / W_{Ai} + \dots + \text{const} \quad \dots \quad (15)$$

機能ブロックにn個のベクトルA<sub>i</sub>が入力され、この機能ブロックからm個のベクトルC<sub>i</sub>が出力される場合\*

$$P = f(A_1, A_2, \dots, A_n, C_1, C_2, \dots, C_m) + \text{const} \quad \dots \quad (16)$$

\* (5)、入力ベクトルAおよびBの他に出力ベクトルCを用いて表した近似式(6)についてそれぞれ説明した。次に、式(4)～(6)にそれぞれ関連する他の近似式について説明する。

【0060】機能ブロックにn個のベクトルA<sub>1</sub>～A<sub>n</sub>が入力される場合には、所定の関数fと所定の定数c<sub>0</sub>、n<sub>s</sub>、t<sub>s</sub>とを用いて、次の(8)を用いて消費電力を算出することもできる。

【0061】

※また、機能ブロックにn個のベクトルA<sub>1</sub>が入力され、入力ベクトルA<sub>i</sub>のビット幅をW<sub>Ai</sub>、ベクトルA<sub>i</sub>のうち所定の論理値にあるもののビット数をH<sub>Ai</sub>、ベクトルA<sub>i</sub>に起因して消費される電力の最大値をP<sub>Ai</sub>とした場合、

★のような一次線形式には限定されず、例えば(12)式のような二次非線形式も含まれる。

【0064】

★3)式のように任意の個数選ばれたベクトルA<sub>i</sub>のみを用いてよい。

【0066】

40用いてよい。

【0067】

◆も含まれる。

【0068】

\*は、次の(16)式を用いてよい。

ここで、(16)式には機能ブロックに入力されるベクトルAの全てと出力されるベクトルCの全てが含まれるが、それぞれ任意の個数選ばれた $\Delta_i$ および $C_i$ のみで\*

$$P = f (A_1, \dots, C_j, \dots) + \text{const} \quad \dots \quad (17)$$

機能ブロックにn個のベクトル $A_i$ が入力され、この機能ブロックからベクトル $C_i$ が出力され、ベクトル $A_i$ のビット幅を $W_{Ai}$ 、ベクトル $A_i$ のうち所定の論理値※

$$P = f (P_{A1}, P_{A2}, \dots, P_{An}, H_{A1}, H_{A2}, \dots, H_{An}, W_{A1}, W_{A2}, \dots, W_{An}, P_{C1}, P_{C2}, \dots, P_{Cm}, H_{C1}, H_{C2}, \dots, H_{Cm}, W_{C1}, W_{C2}, \dots, W_{Cm}) + \text{const} \quad \dots \quad (18)$$

で表わされる近似式を用いてよい。

\*表わされた(17)式を用いてよい。  
【0069】

※にあるもののビット数を $H_{Ci}$ 、ベクトル $C_i$ に起因して消費される電力の最大値を $P_{Ci}$ とした場合、

★る。

【0070】上記(18)式は、次の(19)式のみならず、例えば(20)式のような二次非線形形式も含まれ★

$$P = f = P_{A1} \cdot H_{A1} / W_{A1} + P_{A2} \cdot H_{A2} / W_{A2} + \dots + P_{An} \cdot H_{An} / W_{An} + P_{C1} \cdot H_{C1} / W_{C1} + P_{C2} \cdot H_{C2} / W_{C2} + \dots + P_{Cm} \cdot H_{Cm} / W_{Cm} + \text{const} \quad \dots \quad (19)$$

$$P = f = P_{A1} \cdot H_{A1}^2 / W_{A1} + P_{A2} \cdot H_{A2}^2 / W_{A2} + \dots + P_{An} \cdot H_{An}^2 / W_{An} + P_{C1} \cdot H_{C1}^2 / W_{C1} + P_{C2} \cdot H_{C2}^2 / W_{C2} + \dots + P_{Cm} \cdot H_{Cm}^2 / W_{Cm} + \text{const} \quad \dots \quad (20)$$

ここで、任意の個数選ばれた入力ベクトル $A_i$ および出力ベクトル $C_i$ とを用いて、

$$P = f (P_{A1}, H_{A1}, W_{Ai}, \dots, C_j, H_{Cj}, W_{Cj}) + \text{const} \quad \dots \quad (21)$$

で表わされる近似式を用いてよい。

◆る。

【0072】上記(21)式は、次の(22)式のみならず、例えば(23)式のような2次非線形形式も含まれ◆

$$P = f = P_{Ai} \cdot H_{Ai} / W_{Ai} + \dots + C_j \cdot H_{Cj} / W_{Cj} + \dots + \text{const} \quad \dots \quad (22)$$

$$P = f = P_{Ai} \cdot H_{Ai}^2 / W_{Ai} + \dots + C_j \cdot H_{Cj}^2 / W_{Cj} + \dots + \text{const} \quad \dots \quad (23)$$

次に、機能ブロックにn個のベクトル $A_i$ が入力され、<sup>30\*</sup> ベクトル $A_i$ のビット幅を $W_{Ai}$ 、ベクトル $A_i$ のうち前時刻から現在の時刻へ移行する際に論理値が変化した\*

$$P = f (P_{A1}, P_{A2}, \dots, P_{An}, H_{A1}, H_{A2}, \dots, H_{An}, W_{A1}, W_{A2}, \dots, W_{An}) + \text{const} \quad \dots \quad (24)$$

で表わされる近似式を用いることができる。

※ならず、例えば(26)式も含まれる。

【0074】この(24)式には、次の(25)式のみ※

$$P = f = P_{A1} \cdot H_{A1} / W_{A1} + P_{A2} \cdot H_{A2} / W_{A2} + \dots + P_{An} \cdot H_{An} / W_{An} + \text{const} \quad \dots \quad (25)$$

$$P = f = P_{A1} \cdot H_{A1}^2 / W_{A1} + P_{A2} \cdot H_{A2}^2 / W_{A2} + \dots + P_{An} \cdot H_{An}^2 / W_{An} + \text{const} \quad \dots \quad (26)$$

ここで、任意の個数選ばれた入力ベクトル $A_i$ および出力ベクトル $C_i$ とを用いて

$$P = f (P_{Ai}, H_{Ai}, W_{Ai}) + \text{const} \quad \dots \quad (27)$$

で表わされる近似式を用いてよい。

★らず、例えば(29)式のような2次非線形形式も含む。

【0076】上記(27)式は、次の(28)式のみな☆

$$P = f = P_{Ai} \cdot H_{Ai} / W_{Ai} + \dots + \text{const} \quad \dots \quad (28)$$

$$P = f = P_{Ai} \cdot H_{Ai}^2 / W_{Ai} + \dots + \text{const} \quad \dots \quad (29)$$

機能ブロックにn個のベクトル $A_i$ が入力され、m個のベクトル $C_j$ が出力され、ベクトル $A_i$ のビット幅を $W_{Ai}$ 、ベクトル $A_i$ のうち前時刻から現在の時刻へ移行する際に論理値が変化したもののビット数を $H_{Ai}$ 、ベ

クトル $A_i$ に起因して消費される電力の最大値を $P_{Ai}$ とし、ベクトル $C_j$ のビット幅を $W_{Cj}$ 、ベクトル $C_j$ のうち前時刻から現在の時刻へ移行する際に論理値が変化したもののビット数を $H_{Cj}$ 、ベクトル $C_j$ に起因し

て消費される電力の最大値を  $P_{Ci}$  とし、任意の個数選ばれた  $A_i$  と  $C_i$  とにより表わされる次の(30)式を\* 【0078】

$$P = f - PA_i \cdot HA_i, WA_i, \dots, C_j, HC_j, WC_j, \dots) + \text{const} \quad \dots \quad (30)$$

この(30)式は、次の(31)式のみならず、例えば(32)式のような2次非線形形式も含まれる。※

$$P = f - PA_i, HA_i / WA_i + \dots + C_j, HC_j / WC_j + \dots + \text{const} \quad \dots \quad (31)$$

$$P = f = PA_i \cdot HA_i^2 / WA_i + \dots + C_j, HC_j^2 / WC_j + \dots + \text{const} \quad \dots \quad (32)$$

機能ブロックに  $n$  個のベクトル  $A_i$  が入力される場合、現時刻の  $A_i$  を  $A_i(t)$ 、前時刻の  $A_i$  を  $A_i(t-1)$  としたときに、次の(33)式を用いてよい。

$$P = (A_1(t), A_1(t-1), A_2(t), A_2(t-1), \dots, A_n(t), A_n(t-1)) + \text{const} \quad \dots \quad (33)$$

あるいは、機能ブロックに  $n$  個のベクトル  $A_i$  が入力され、 $m$  個のベクトル  $C_j$  が川力される場合、現時刻の  $A_i$  を  $A_i(t)$ 、前時刻の  $A_i$  を  $A_i(t-1)$  とし、☆ 【0080】

$$P = f (A_1(t), A_1(t-1), A_2(t), A_2(t-1), \dots, A_n(t), A_n(t-1), C_1(t), C_1(t-1), C_2(t), C_2(t-1), \dots, C_m(t), C_m(t-1)) + \text{const} \quad \dots \quad (33)$$

以上のように、本実施例によればアーキテクチャ設計工程のような上位の設計工程において消費電力を算出することができる。このため、論理回路設計やトランジスタ回路設計といった下位の設計工程まで進まないと消費電力を見積もることができなかつた従来の場合と比較し、消費電力の最適化が可能であり、かつ設計時間及びコストの低減が可能である。

#### 【0082】

【発明の効果】以上説明したように、本発明の集積回路の消費電力算出方法は、動作に必要な機能ブロックがいずれであるかを検出し、検出した機能ブロックがそれぞれ消費する電力を合計していくことで回路全体の消費電力を求めるものであり、回路を複数の機能ブロックで構成する上位の設計工程において消費電力を算出することができるため、消費電力の最適化が可能でかつ設計時間及びコストの低減が可能である。

【0083】このような本発明の消費電力算出方法は、本発明の消費電力算出装置を用いて実施することができる。

#### 【図面の簡単な説明】

【図1】本発明の一実施例による集積回路の消費電力算出装置の構成を示したブロック図。

【図2】本発明の一実施例による集積回路の消費電力算出方法における処理の手順を示したフローチャート。

【図3】同消費電力算出方法を適用して消費電力を求めることができ可能な集積回路としてマイクロプロセッサを例にとり、そのアーキテクチャ設計工程のモデルを示した

\*用いることもできる。

$$P = f - PA_i \cdot HA_i, WA_i, \dots, C_j, HC_j, WC_j, \dots) + \text{const} \quad \dots \quad (30)$$

$$P = f - PA_i, HA_i / WA_i + \dots + C_j, HC_j / WC_j + \dots + \text{const} \quad \dots \quad (31)$$

$$P = f = PA_i \cdot HA_i^2 / WA_i + \dots + C_j, HC_j^2 / WC_j + \dots + \text{const} \quad \dots \quad (32)$$

★1) としたときに、次の(33)式を用いてよい。

$$P = (A_1(t), A_1(t-1), A_2(t), A_2(t-1), \dots, A_n(t), A_n(t-1)) + \text{const} \quad \dots \quad (33)$$

☆現時刻の  $C_j$  を  $C_j(t)$ 、前時刻の  $C_j$  を  $C_j(t-1)$  としたときに、次の(34)式を用いてよい。

$$P = f (A_1(t), A_1(t-1), A_2(t), A_2(t-1), \dots, A_n(t), A_n(t-1), C_1(t), C_1(t-1), C_2(t), C_2(t-1), \dots, C_m(t), C_m(t-1)) + \text{const} \quad \dots \quad (33)$$

回路図。

【図4】論理回路の構成の一例を示した回路図。

【図5】トランジスタ回路の構成の一例を示した回路図。

【図6】本発明の一実施例による集積回路の消費電力算出方法を用いて消費電力を求めた結果を示したグラフ。

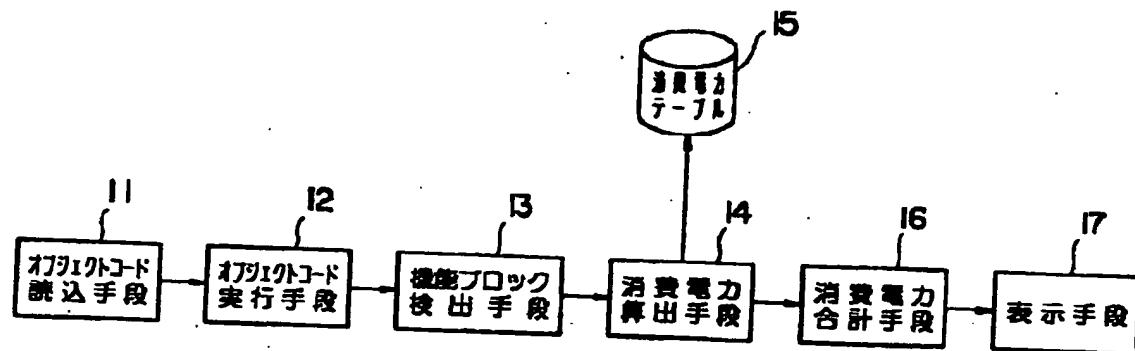
#### 【符号の説明】

- 1 1 オブジェクトコード読込手段
- 1 2 オブジェクトコード実行手段
- 30 1 3 機能ブロック検出手段
- 1 4 消費電力算出手段
- 1 5 消費電力テーブル
- 1 6 消費電力合計手段
- 1 7 表示手段
- 2 1 プログラムカウンタ
- 2 2 命令キャッシュ
- 2 3 インクリメンタ
- 2 4, 2 9, 3 4 マルチプレクサ
- 2 5, 2 7, 3 1, 3 3 ラップ
- 40 2 6 レジスタファイル
- 2 8 ブランチアドレス生成器
- 3 0 算術論理演算器
- 3 2 データキャッシュ
- 4 1, 4 2 NANDゲート
- 5 1, 5 2 Pチャネルトランジスタ
- 5 3, 5 4 Nチャネルトランジスタ

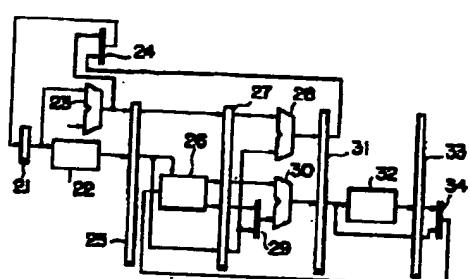
(11)

特開平8-44788

【図1】



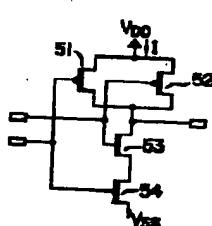
【図3】



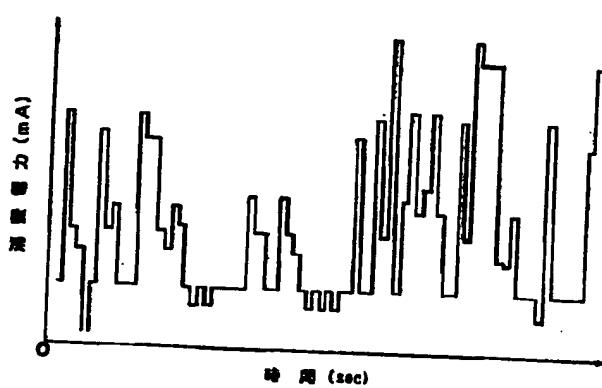
【図4】



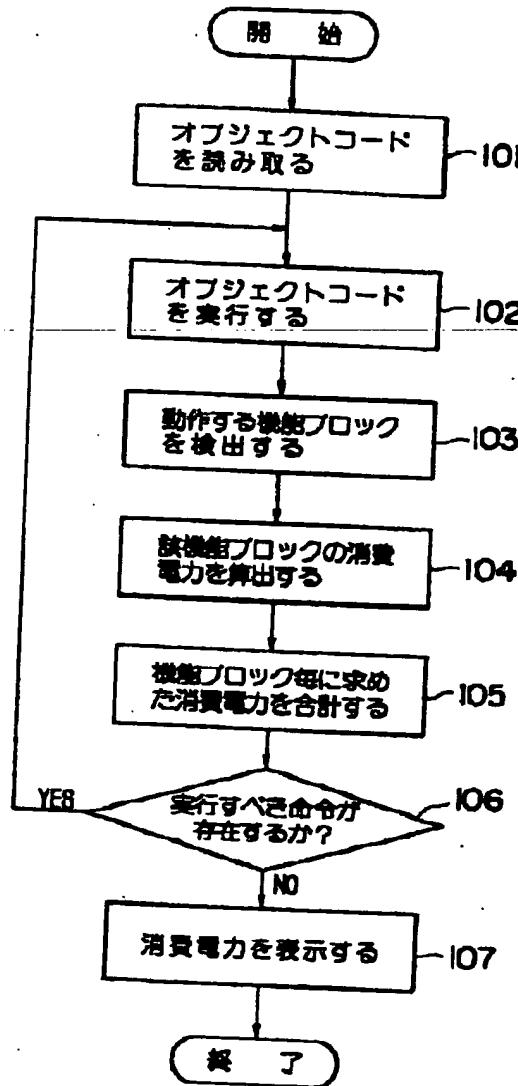
【図5】



【図6】



【図2】



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**